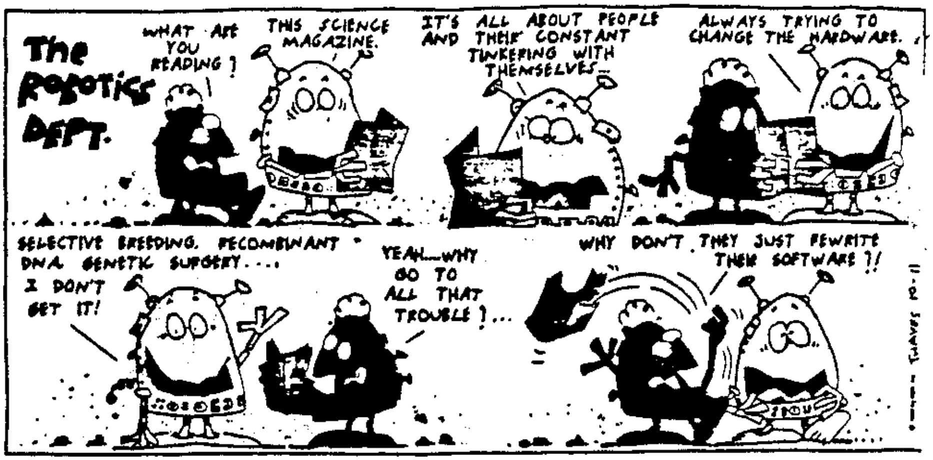
Newsletter Nine-T-Nine
November 1992 Issue

Texas Instruments
USERS GROUP
TORONTO

FOR THE TI-99/4A COMPUTER


The Robotics Dept.
WHAT ARE YOU READING?
THIS SCIENCE MAGAZINE.
IT'S ALL ABOUT PEOPLE AND THEIR CONSTANT TINKERING WITH THEMSELVES...
ALWAYS TRYING TO CHANGE THE HARDWARE.
SELECTIVE BREEDING, RECOMBINANT DNA, GENETIC SURGERY....
I DON'T GET IT!
YEAH...WHY GO TO ALL THAT TROUBLE?...
WHY DON'T THEY JUST REWRITE THEIR SOFTWARE?!
THAVES 12-11

FROM:
9T9 USERS GROUP
15 KERSDALE AVE.
TORONTO,ONT.,M6M-1C9
CANADA

To:

Canada 125

9T9 Users Group

''*When you said you were assembling a clone I thought you wre talking about a computer.*''

9T9 USERS GROUP EXECUTIVE COMMITTEE

PRESIDENT Steve Mickelson (657-1494)
VICE-PRESIDENT Neil Allen (236-0842)
SECRETARY/MEMBERSHIPS Randy Rossetto (469-3468)
TREASURER/OFFICER AT LARGE Cecil Chin (671-2052)

LIBRARY DIRECTORS

Gary Bowser (960-0925)
Andy Parkinson (275-4427)
Steve Findlay (416) 727-6807

NEWSLETTER EDITOR

Steve Mickelson (657-1494)

MEMBERSHIP FEE'S

FULL MEMBERSHIP...................... $30.00 / year
NEWSLETTER SUBSCRIPTION .... $20.00 / year

All memberships are household memberships. A newsletter subscription is only for those who do not wish to attend meeting, but wish to receive our newsletter and have access to our library. You are welcome to visit one of our general meetings before joining the group. If you wish more information contact either our president, in writing, at the club address on the front cover or by phone.

The meetings are usually held on the last Wednesday of each month,(exceptions are December's meeting date, usually mid-month and the months of July and August, when there are no meetings. Consult this issue of Newsletter 9T9 for the date and time of the next meeting. Meetings are usually held at Neil Allen's place, 52 Graystone Gardens,south of Bloor St., just west of Islington Ave., at. 7:30 P.M. from 7:30 - 10:30 PM.

BBS

The 9T9 Users Group supports the Toronto BBS, The TI Tower BBS #(416) 921-2731, 300/1200/2400 BPS, 24 hrs. Sysop, Gary Bowser.

MAILING ADDRESS:

9T9 Users Group, 15 Kersdale Ave., Toronto, Ontario, M6M 1C9, Canada

COMMERCIAL ADVERTISING

Any business wishing to reach our membership may advertise in our newsletter.
The rates are as follows. (width by height):

FULL PAGE	(7" x 10")	$30.00
HALF PAGE	(7" x 5")	$15.00
QUARTER PAGE	(7" x 2 1/2")	$ 7.50

Please have your ad's camera ready and paid for in advance. For more information contact the editor. Don't forget, that any member wishing to place ad's, may do so free of charge as long as they are not involved in a commercial enterprise.

NEWSLETTER ARTICLES

Members are encouraged to contribute to the newsletter in the form of articles, mini programs, helpful tips, hardware modifications, jokes, cartoons and questions. Any article may be submitted in any form by mail or modem. We welcome the reprinting of any article appearing in this newsletter providing credit is given to the author and 9T9. If more information is required, call the editor. The names, 9T9, Nine-T-Nine, Newsletter 9T9, 9T9 Users Group, and Nine-T-Nine Users Group are Copyright, (c) ,1982-1992, by the 9T9 Users Group of Toronto, Canada, all rights reserved.

DISCLAIMER

Opinions expressed in this newsletter are those of the writers and are not necessarily those of the 9T9 USERS' GROUP. 9T9 cannot assume liability for errors or omissions in articles, programs or advertisements. Any hardware modification or project is presented for informational purposes, and the author, newsletter editor, staff and/or 9T9 Users Group &cannot be held liable for any damage to the user's equipment. All such projects are done at your own risk!

©Creative Computing

TIDBITS #63

-By Steve Mickelson, President 9T9 Users Group
Compuserve 76545,1255; Delphi SMICKELSON; GEnie S.MICKELSON

New Editor:
I feel confident someone will step-in as newsletter editor and take-on responsibilities. Some fresh ideas would greatly energize the club, as, other than the Assembly Language S.I.G. and our Newsletter, most activities are at a low ebb. Perhaps a new president, will be able to get things back on track.

Reference Gem:
While recently browsing a local used book store, I found a reference book, that I don't recall being mentioned in other newsletter. The books title:

OSBORNE 16-Bit Microprocessor Handbook
-by Adam Osborne and Gerry Kane

published by Osborne/McGraw-Hill ISBN 0-931988-43-8

This reference has an interesting chapter on the Texas Instruments TMS 9900, TMS 9980 and TMS 9940 microprocessor chips. The TMS 9900 Microprcessor is described with text and diagram of the functional overview; charts of the features of the various programmable registers/memory addresses, TMS 9900 status flags, pin-outs, Instruction and timing.

Also covered are the TMS 9980A and 9981 Microprocessors, the 9940 single-chip "Microcomputer", TIM 9904 Four-Phase clock generator/driver, TMS 9901 PSI (Programmble system interface),TMS 9902 asynchronous communications controller, and the TMS 9903 Synchronous communication controller. Data shhets for the above are detaled, as well.

This book has a wealth of information and reference material, well-suited to the hardware hacker, software writer or repair rep.

I have purchased an extra copy, for $3.00, for the club library. If others areinterested in this text, a visit to your local reference library may be a source of this book.

Well that's it for this month's TId Bits, remember nominations for 1993, come at our mid-December meeting!

Wheelers and Dealers In TI Land:
In a recent editorial, Chris Bobbitt of Asgard Software, ascribed many of our communities woes, to the failure, or should we say lack of success, of the Myarc 9640.

While, it is true, that many talented programmers bought 9640's, and began programming for the computer, the problem wasn't the failures of Myarc Inc., but the limitations of the TI.

Don't forget, that Myarc did bring many hardware upgrades, that helped extend the usefulness of the TI into the 80's and 90's; more speed, memory, 80 column display, full size keyboard, improved BASIC interpreter, hard drive controllers and TI compatibility. Even now, there is a movement to try to emulate the TI operating system on an IBM clone, a case of re-inventing the wheel, as you require an expensive 486 AT-class,(or better), system to get TI emulation at 9640 speeds. Why Tiers wouldn't buy a 9640, at its original price, but will consider paying ten times as much on a super AT to emulate a TI, illustrates part of the problem in TI Land. TI users refuse to support small, independent manufactures, instead going with large commercial companies.

In addition to setting the hardware standards for our community, Myarc recognized, it is far more practical to perfect the wheel, than to re-invent same. The Press, was a case where an ambitious programmer sought to re-invent word processing for the TI community. It was a dismal failure.

As with their hardware, Myarc considered to take what existed, advance it one step farther, and keep compatibility. The HFDC, 9938 display chip, 9995 CPU etc., all allowed backward compatibility with the TI-99/4A, yet improved the overall operation of the software.

Though poorly managed and marketed, Lou Phillips and his Myarc Computer company contributed much to a community. The fact that many good programmers purchased and started programming for the Geneve, illustrates the relative lack of good integrated hardware upgrades. The Mechatronic 80-column, Digit AVPC 80-column and TIM 80-column card are three examples of how the community lacked either the money or interest, to rally behind a single product. As a result, including the 9640, we now have four different 80-column devices. The TIM may well become the standard, only because it is still in production

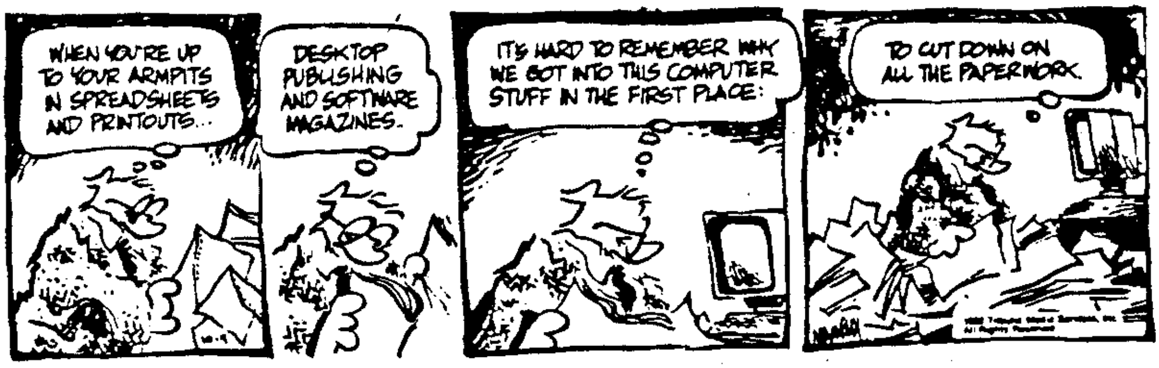
As for keyboard upgrades, there was one from Kansas(a kit), a joystick interface from Germany,(a kit also), the 9640, and the RAVE interface (both assembled or kit). The RAVE unit may eventually win-out, for the same reasons as the TIM.

I could continue on, with comparisons of RAM disks, GRAM devices and GPL interpreters, clock cards, mouse interfaces, RAM cards, new forms of BASIC interpreters, speed upgrades, CPU upgrades or accelerator cards, etc.,etc., all were integrated into the Geneve.

I feel the current, and much of our future hardware design standards, were/will be set by Philips, in his 9640. We owe much to Lou Philips, for these and other improvements to our TI computer system. Myarc may be blamed for the TI community's problems, but many of those problems, were created by lack of support from our community.

Much of the failure can be attributed to an apathetic community, used to cheap,(price not quality), software and hardware. Many current users, acquired their systems and software, for next to nothing, and are not interested in improving their TI, if it will cost anything. We should thank those who continue to contribute to our community. Their support, stems from love of the TI and the community itself.

So next year, make a promise to both yourself and the community, in general, by providing financial support to those who still market products for the TI, as they don't have the financial resources to support research. Perhaps User Groups should increase membership dues by $5.00 and use that money to give support to the commercial sources, And help keep a good thought!


WHEN YOU'RE UP TO YOUR ARMPITS IN SPREADSHEETS AND PRINTOUTS...
DESKTOP PUBLISHING AND SOFTWARE MAGAZINES.
IT'S HARD TO REMEMBER WHY WE GOT INTO THIS COMPUTER STUFF IN THE FIRST PLACE:
TO CUT DOWN ON ALL THE PAPERWORK.

DON'T QUOTE ME, BUT! **by 9T9 Assembly SIG Group**
SEPT./92

Over the past four months, the group has been working on, (and still is), a text file viewer. It will be the first program which can view a file of any size, and will have the ability to scroll back to the first line. Features include, auto-detect of V9938 and V9958 video chips for 80 column display, cataloging all drive types, and displaying only DV/80 file names in a selection box. The user can also decide which sections of the file are to be sent to the printer.

As like other SIG programs, 'file viewer' is made up of a number of fully documented, self-contained sub-programs, which may also be used in future programs.

The main 'file viewer' sub-program, with its superior scrolling function, could be used to replace currently limited viewing functions in programs like disk managers, ramdisk menus, etc.

The auto-detect feature of the V9938/58 sub-program is just one of many new VDP routines the SIG has written, which makes the job of writing a program with screen output to 32/40/80 column modes a breeze for the Assembly programmer.

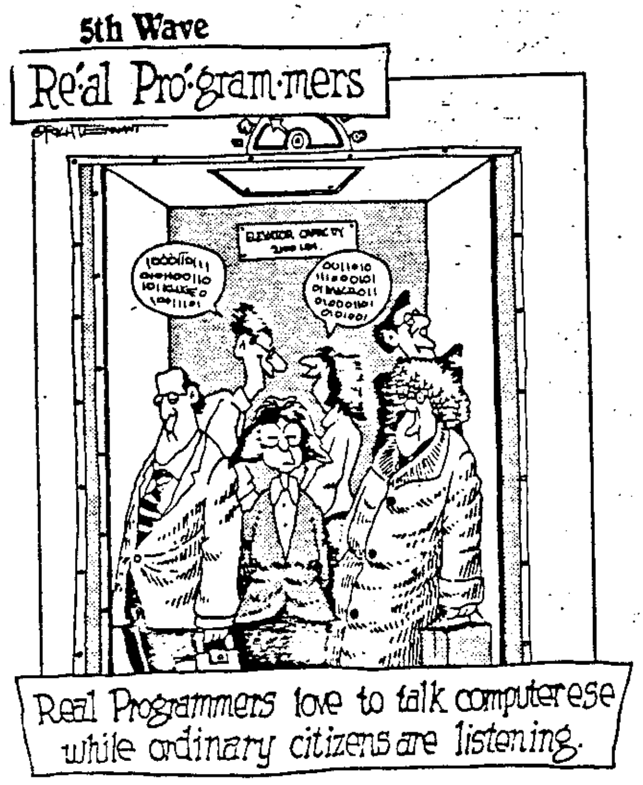
The drive catalog/file selection sub-program allows any future program to replace the normal filename prompt with a selection box that displays only file names with filetype(s) the program selects. This makes it easier for the user, as he/she does not have to remember the filename. It also prevents the user from selecting a file with an incompatible filetype.

The original video chip in a TI console has a maximum screen size of 24 lines of text/graphics, whereas the new V9938/58 chips have a maximum screen size of 27 lines, (really 26 lines, with bottom or top half of a line), in its new 80 column text and graphics modes. Do you know that you can have 27 lines in the original 32 column and multicolor modes, but do you also know WHY you CAN'T have 27 lines in 40 column mode?

If you have any questions for our Assembly SIG members, or want more information on a certain problem, or subject, or source code for one of our sub-programs, or the finished program itself, please write to the 9T9 Users Group.

We are willing to help you solve programming problems, or to help you write a sub-program for a certain task. Send a disk, either blank or with your problematic source code, to the 9T9 Users Group.

Until next month, happy programming - "THE ASSEMBLY SIG"



Real Programmers love to talk computerese while ordinary citizens are listening.

ELECTRONICS NEWS

▶ TI Windfall

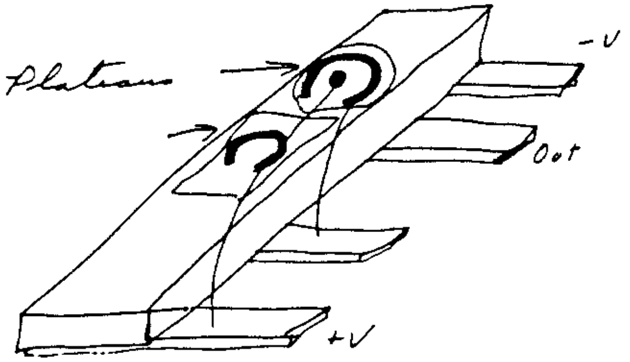
Jack Kilby's original microchip, shown here, is raking in big bucks for Texas Instruments, now that Japan's granted a patent almost thirty years after the first application was made.

Just over a decade of royalties could net TI a cool $10 billion.

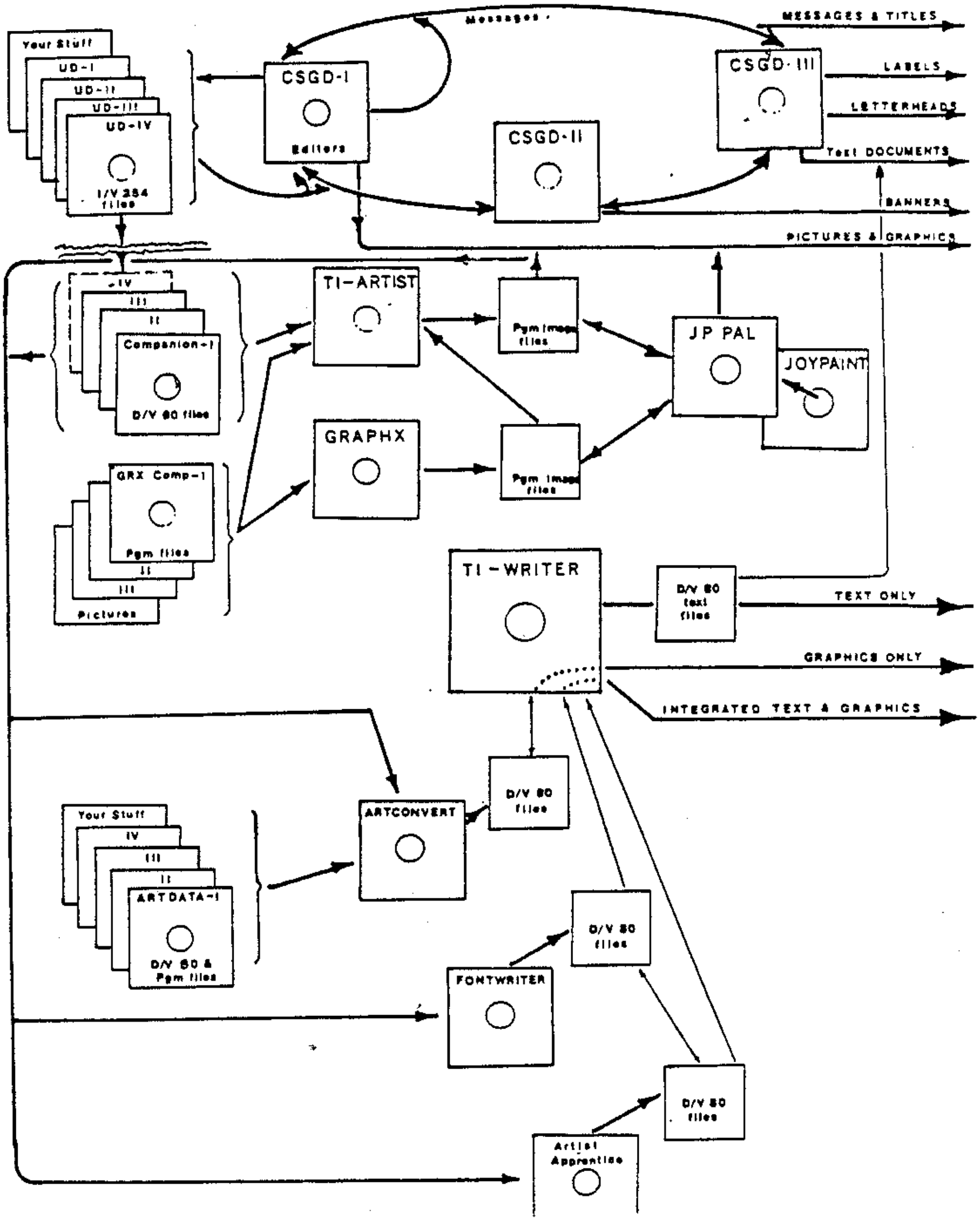
Applications to patent the device in the US and Japan were made in 1960 after Kilby, then a fledgling company engineer with TI, tested the first microchip in October 1958. The US granted a patent in 1964 but the Japanese had not done so until late last year.

Royalties in the first year from Japanese microchip production is expected to exceed TI's annual net income, kicking off at the $400 million mark and growing at a rate of 25% per year until 2001 when the patent expires.

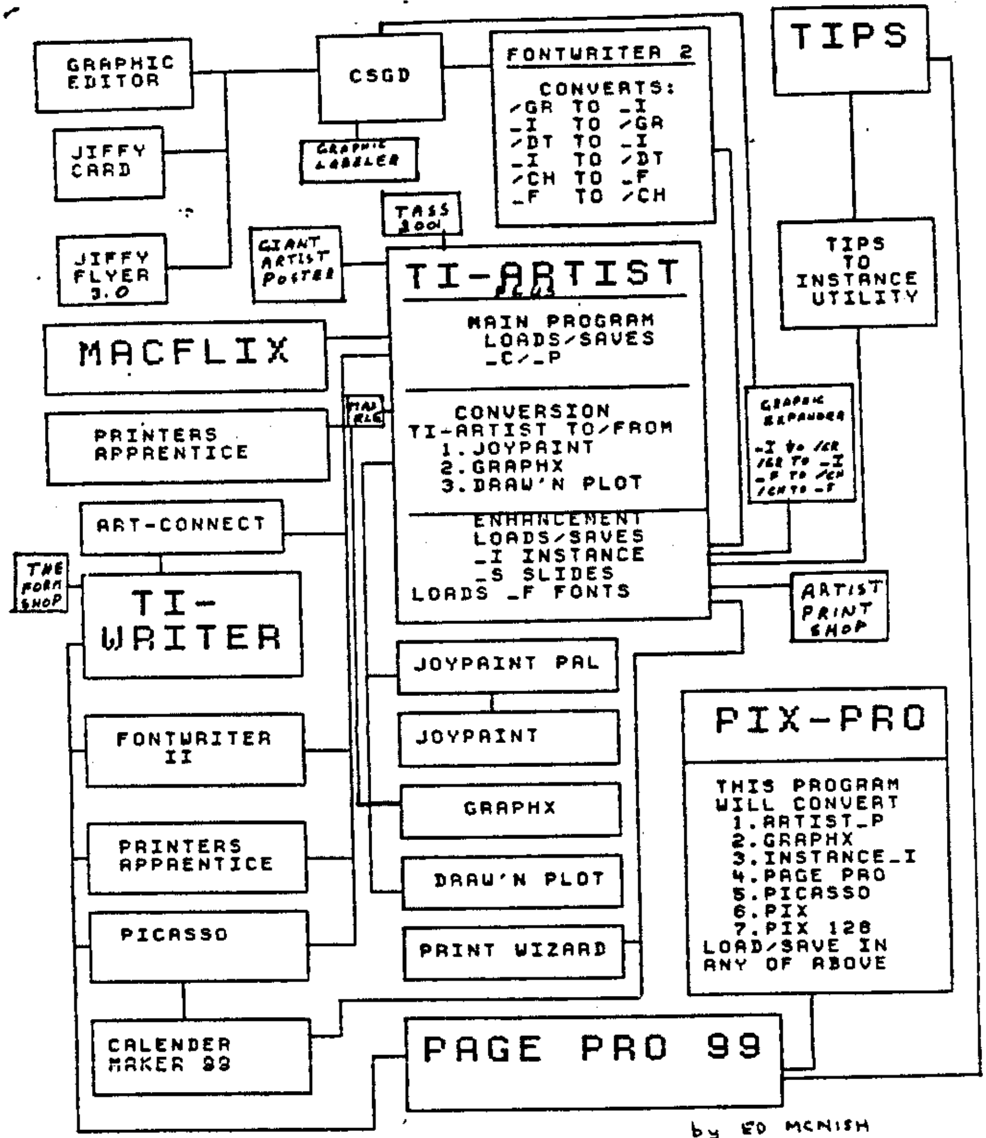
EO NO. 043601
DATE Sept 12, 1958



SOME POPULAR GRAPHICS PROGRAMS



"LIBRARY CORNER"



by ED MCNISH
created with PAGE PRO

Crystal Software to sell MIDI Master 99

MIDI Master 99 will now be marketed by Crystal Software Inc., according to Mike Maksimik, creator of the device.

Previous plans for Asgard Software to market the device have been cancelled.

MIDI Master 99 v2.2, now available, allows loading of disk-based music files and compiling of symbolic music files for playing on the MIDI interface. Version 3.0, which was tentatively scheduled for release toward the end of May, will record keyboard-based music. Maksimik says purchasers of v2.2 will receive v3.0 at no charge as long as they send in the registration card packed with the software.

MIDI Master 99 v2.2, which includes interface, software, two connecting cables and documentation is available for $45 from Michael J. Maksimik, Crystal Software Project, 635 Mackinaw Ave., Calumet City, IL 60409-4014. Checks should be made payable to Michael Maksimik.

TIsHUG NEWS DIGEST September 1992

Writing in Machine Code

The Video Processor, part 1 by J.E. Banfield

The TI99/4A relies on the TMS9918A series of video processor chips, which differ mainly in their video output specifications, which need not concern us. This processor is a very complicated integrated circuit, probably much more complicated than the TMS9900 CPU chip, at least its data manual is three times as thick as the TMS9900 manual.

The video processor controls a 16K dynamic RAM which it uses for screen display data, the colour table and sprite definitions. There is sufficient dynamic RAM address space left over to be a very significant addition to other TI99/4A RAM; the BASIC interpreter uses it extensively. I use it as a buffer for disk data. For example, a full disk track can be set in VDP RAM prior to transfer to a DMA memory in my disk controller.

The control levels required to interface the video processor are defined in Table 6-1, extracted from the data manual. We can now face the job of screen control.

Screen Border Control

The border screen colour is controlled by the rightmost 4 bits of register 7 in the VDP chip. To change this colour code, we need to write to that register, 1 for black, F for white, etcetera.

This write to VDP register can be done in various ways. I chose here a long method which is less complicated than the alternative described later. The method of writing a program using the MiniMemory Easybug option was detailed in the last article and will not be repeated here. Enter the following program starting at M7FC0.

```
M7FC0   02 01    MOVEI 1,
        01 00    "black"
        02 02    MOVEI 2,
        87 00    VDP register 7
        D8 01    MOVB @.+2, 1
        8C 02    VDP write address
        10 00    SKIPA nil (delay)
        D8 02    MOVB @.+2, 2
        8C 02    vdp write address
        10 FF    SKIPA minus 1 (STOP)
```

Note that the first byte written contains the data, the second byte specifies a write to register 7 (see Table 6-1). Now press "." and type "E7FC0<enter>". The border colour goes black and we enter an infinite loop. Turn off the console and back on again (or reset any way you can).

Let us examine the program.

```
M7FC0   02 01
0 0 0 0  0 0 1 0  0 0 0 0  0 0 0 1
\_______020_____________/ \___________/
    op-code               S address
    MOVEI                   Ac 1
```

See Figure 1 in the last article. The instruction moves the immediate data, to be found in the next word (01 00) to Ac 1.

```
M7FC4   02 02
0 0 0 0  0 0 1 0  0 0 0 0  0 0 1 0
\_______020_____________/ \___________/
    op-code               S address
    MOVEI                   Ac 2
```

The data in the next word (87 00) is placed in Ac 2.

```
M7FC8   D8 01
1 1 0 1  1 0 0 0  0 0 0 0  0 0 0 1
\_____/ \_____________/ \___________/
op-code   D address       S address
 MOVB      @.+2             Ac 1
```

Move the contents of the left hand byte in Ac 1 (that is 01) to the address given in the next word, which is 8C 02.

```
        1 0 0 0  1 1 0 0  0 0 0 0  0 0 1 0
```

This has the address line of A14 high which enables the VDP chip for writing to as can be seen from Figure 6-1 and Table 6-1, that is U100 pin 14 (CSW(L)) will be low and pin 13 (MODE) will be high.

The next instruction is a delay which is required by the Video Processor chip. The following instruction is similar to 7FC8 but specifying the source as the left most byte of Ac 2, so the second byte written is 87, defining VDP register 7. The final instruction, SKIPA minus 1, as explained in the second last article, enters an infinite loop, in effect causing a stop.

OK, if all goes well we will make a change to avoid the inconvenience of the infinite loop. Change the instruction at 7FD2. In Easybug type:

```
M7FD2   04 60    JUMPA @.+2
        7F E0    address 7FE0
```

This instruction is made from:

```
M7FD2   04 60
0 0 0 0  0 1 0 0  0 1 1 0  0 0 0 1
\_______044_____________/ \___________/
    op-code               S address
    JUMPA                   @.+2
```

Then enter a short program at 7FE0.

```
M7FE0   02 06    MOVEI 6,
        70 E5    GROM address
        04 60    JUMPA @.+2
        00 60    ROM address
```

This loads Ac 6 with a GROM address (miniMemory GROM entry?) and the next jumps to the GPL interpreter in the console ROM. Check this by executing 7FC0.

There is no change to the border colour but "?" is displayed. Now change the colour data by:

```
M7FC2   09
```

and repeat the execution of 7FC0.

Now examine the contents of M83FE.

```
M83FE = 8C
M83FF = 02
```

Incredible, the VDP write data address is in console RAM and we did not put it there!

Sacred Sites

TI, in its wisdom (??), have sequestrated certain addresses for specific purposes and you may only change their contents AT YOUR PERIL. In particular:

83FA	(Ac D or R13 in GPL interpreter)
83FC	(Ac E or R14 in GPL interpreter)
83FE	(Ac F or R15 in GPL interpreter)

Although we must not change the contents of these registers, there is no bar to using them as data as described for Ac F later.

Multiple VDP Byte Write

Time and space is running out fast so I will leave detailed explanation of these routines to a later article. However, you might like to try them out and analyse the code. This routine writes a selected byte a number of times as defined in the data to incrementing VDP addresses. I use it to set up track data for formatting disks.

```
M7C60   C0 FB    MOVE 3, @ 8+
        C1 03    MOVE 4, 3
        02 43    ANDI 3,
        FF 00
        02 44    ANDI 4,
        00 FF    count
        D8 03    MOVEB @.+2, 3
        8C 00    VDP write data
        06 04    SOS 4
        16 FC    SKIPNE minus 4
        04 58    JUMPA @ 8 (return)
```

This is called by:

```
   06 A0    JSP @.+2
   7C 60
   WX YZ    data; WX is the ASCII byte to be
             written YZ times.
```

Before executing the 7FC0 subroutine, it is necessary to set the VDP address which can be done in the following subroutine.

```
M7A00   C0 7B    MOVE 1, @ B+
        D7 E0    MOVEB @ F, @.+2
        83 E3    Ac 1, right byte
        10 00    delay
        D7 C1    MOVEB @ F, 1
        04 5B    JUMPA @ B
```

In fact, we can use the 7A00 subroutin to change the border colour as as alternative to the 7FC0 routine.

Now enter the program:

```
M7A40   06 A0    JSP @.+2
        7A 00    set VDP address
        40 E3    address with write flag
        06 A0    JSP @.+2
        7C 60    multiple byte write
        41 10    write "A" 16 times
        04 60    JUMPA @.+2
        7F E0    to ? in Easybug
```

Now execute 7A40 several times. Try changes to address, data and count.

From Table 6-1, you will see that the only difference in VDP register write from VDP address write, is in the second byte written. So try:

```
M7A20   06 A0    JSP @.+2
        7A 00    set VDP address
        87 01    or 87 09 or 87 0X
        04 60    JUMPA @.+2
        7F E0    to ? in Easybug
```

The next article in this series will explain the above and show how to write the colour table in VDP RAM.

Correction to the article in Volume 11 Number 5, page 20. "Not valid for TMS9900" should read "Not valid for the TI99/4A". The TI99/4A lacks address decoding needed to implement these TMS9900 instructions.

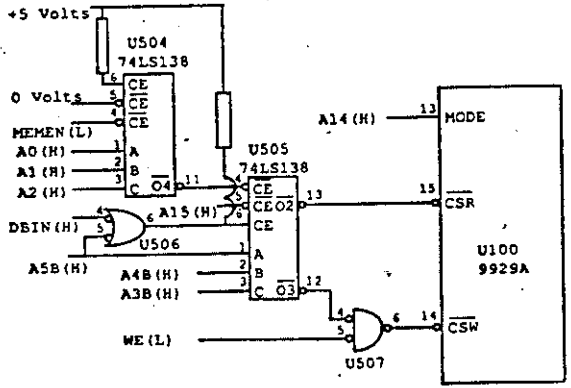


Figure 6-1
From TI99/4A Schematics Diagrams, sheets 1 and 4

$O4[U504](L)=MEMEN\cdot A0\cdot\overline{A1}\cdot\overline{A2}$

$O2[U505](L)=O4[U504]\cdot\overline{A15}\cdot(\overline{DBIN}+\overline{A5B})\cdot\overline{A5B}\cdot A4B\cdot\overline{A3B}$

$CSR(L)=MEMEN\cdot A0\cdot\overline{A1}\cdot\overline{A2}\cdot\overline{A3B}\cdot A4B\cdot\overline{A5B}\cdot\overline{A15}$

```
x=don't care        1000  10xx  xxxx  xxx0
                      8     8           even
```

$O3[U505](L)=O4[U504]\cdot\overline{A15}\cdot(\overline{DBIN}+\overline{A5B})\cdot A5B\cdot A4B\cdot\overline{A3B}$

$CSW(L)=WE\cdot O3[U505]$
$=WE\cdot\overline{DBIN}\cdot MEMEN\cdot A0\cdot\overline{A1}\cdot\overline{A2}\cdot\overline{A3B}\cdot A4B\cdot A5B\cdot\overline{A15}$

```
                         1000  11xx  xxxx  xxx0
                           8     C           even
```

OPERATION	BIT 0	1	2	3	4	5	6	7	$\overline{CSW}$	$\overline{CSR}$	MODE
WRITE TO VDP REGISTER											
BYTE 1 DATA WRITE	D_0	D_1	D_2	D_3	D_4	D_5	D_6	D_7	0	1	1
BYTE 2 REGISTER SELECT	1	0	0	0	0	RS_0	RS_1	RS_2	0	1	1
WRITE TO VRAM											
BYTE 1 ADDRESS SETUP	A_6	A_7	A_8	A_9	A_{10}	A_{11}	A_{12}	A_{13}	0	1	1
BYTE 2 ADDRESS SETUP	0	1	A_0	A_1	A_2	A_3	A_4	A_5	0	1	1
BYTE 3 DATA WRITE	D_0	D_1	D_2	D_3	D_4	D_5	D_6	D_7	0	1	0
READ FROM VDP REGISTER											
BYTE 1 DATA READ	D_0	D_1	D_2	D_3	D_4	D_5	D_6	D_7	1	0	1
READ FROM VRAM											
BYTE 1 ADDRESS SETUP	A_6	A_7	A_8	A_9	A_{10}	A_{11}	A_{12}	A_{13}	0	1	1
BYTE 2 ADDRESS SETUP	0	0	A_0	A_1	A_2	A_3	A_4	A_5	0	1	1
BYTE 3 DATA READ	D_0	D_1	D_2	D_3	D_4	D_5	D_6	D_7	1	0	0

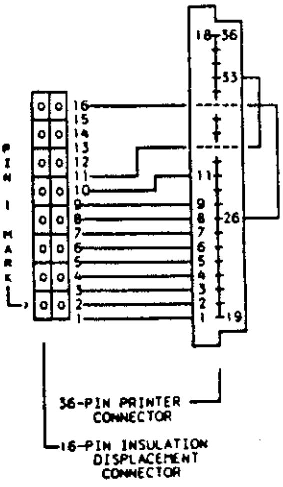
Table 6-1 - CPU/VDP Data Transfers. From the TMS9918A Manual

CIN-DAY

TI 99/4A PARALLEL PRINTER CABLE

Last week I helped one of our members make up a parallel printer cable for a new printer, and it turned out to be so easy that I decided to share it so you can make up a spare for your system. The easiest way is to use some insulation-displacement connectors and ribbon cable you don't have to do any soldering, but you can use hooded connectors and 12-conductor cable to make a fancier hookup. If you use sixteen-conductor ribbon cable, go for the multi-colored type since it's a lot easier to keep track of the lines. Six feet of it cost about $3. The 16-pin female IDC connector is about $2 and the 36-pin "Centronics" male connector is about $4. All you have to do is clamp the 16-conductor ribbon in the 16-pin connector on one end and split the wires on the other end so you can seperate the 3 that don't go in order Notice that the numbering scheme is different on the opposite end of the cable, but the #1 pin is marked with an arrow on the computer end and the printer end is marked on pins 1 & 19 and pins 18 & 36. This means that on the printer end with an IDC connector you use every other wire slot for 1-9. Be sure the wire is firmly inserted in each terminal pin so the insulation is displaced (hence the name, insulation displacement connector). Or you can use 36-connector cable and use every other wire on the 16-pin plug end for 1-9. If you have any problems, give me a call at 714/628-6886.

Kawartha Kronicle Vol 7 No 4 Gene Bohot

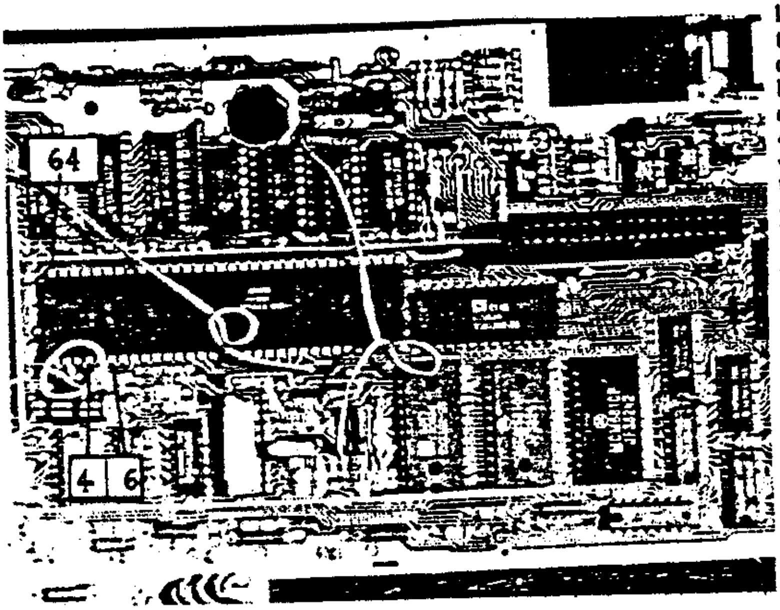


ip
ed

ide

Pause, Quit, or Interrupt

Three buttonswitches could have been very useful on the 4A if TI had put them there; but they probably didn't think the 99 would get this far. The three are: A pause switch, a reset or quit button, and a load interrupt switch. Some of us have the load interrupt switch sold by Tenex; but it requires a separate piece to be inserted in the already cumbersome "foot" construction, and is unwieldy. Here is a simple modification to install all three in one sweep:

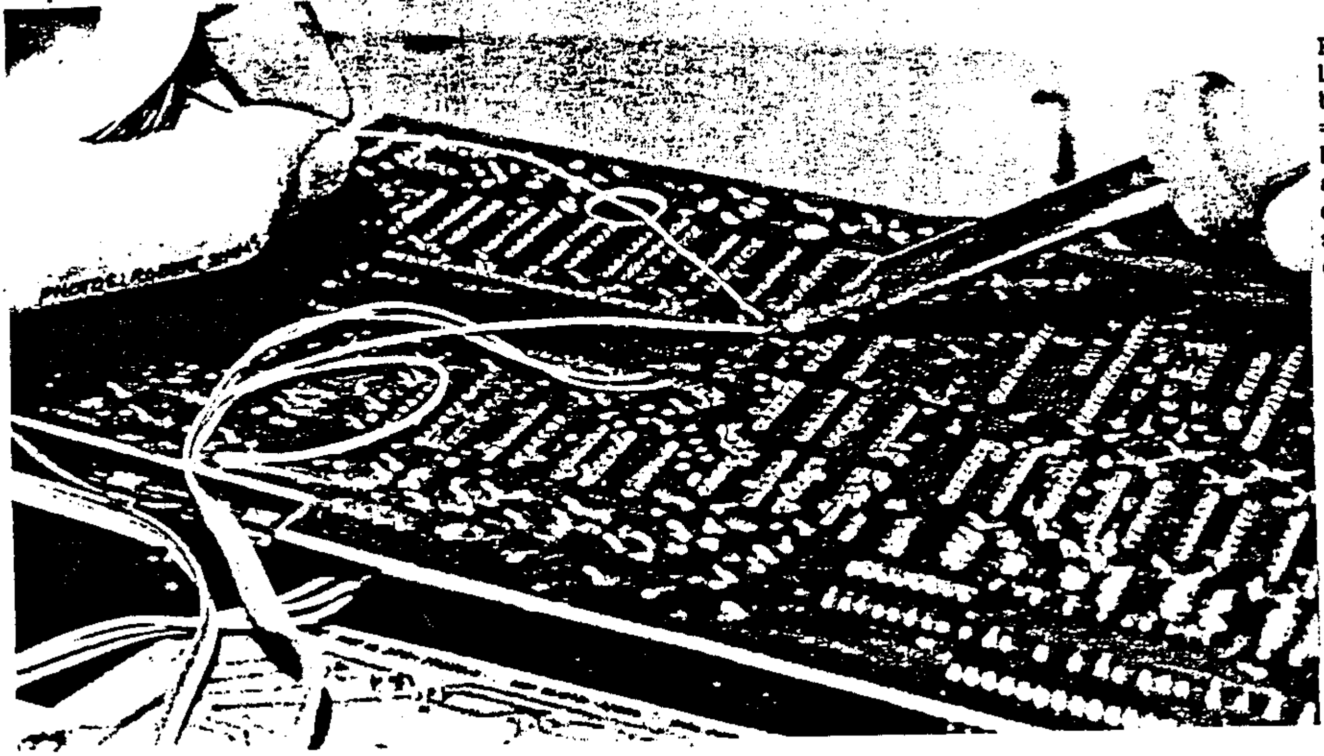


STEP 1: Locate the main processing chip (CPU): it is the largest chip on the board. Looking at the chip from the top (component side), identify pins 4, 6, ad 64. #64 will be on the side of the CPU nearest the large hole in the board, and farthest from the GROM port socket (see drawing at left). #1 is directly across the chip from it; so 4 and 6 are the 4th and 6th pins to the right.(All this is very elemental unless you have never worked with chips before.) Now, turn the board over and identify these same pins from the back (solder side). Don't get confused!

STEP 2: Get four pieces of stranded wire about a foot long, or a section of ribbon cable of that length and containing four wires. Separate about two inches of the wires on each end of the cable. Strip the ends of the wires about 1/4 inch, and tin them (heat with solder gun and touch solder to the wire to make it stiff). With needle nose pliers or tweezers, bend each lead into a tight curl. Put one end of three of the wires over the protruding pins of 4, 6, and 64, and deftly solder each in place. Be sure to use a low wattage iron, with a sharply point tip, fully preheated, so that you can solder the leads within a second or so. Now solder one end of the fourth wire to a nearby strip of the ground of the board, such as that which runs around the outside of the board.

STEP 3: Secure two switches: a miniature, momentary pushbutton, and an three position toggle switch, normally open, momentary one direction and closed the other. If you cannot find the toggle switch with one side momentary, get one with both sides locked. Mount the two switches on the console in a place that will be convenient without being dangerous (I put mine on the right side, right up next to the front of the speech systhesizer).

STEP 4: Connect the opposite end of the wire from pin 6 to one side of the momentary pushbutton. This is a reset switch. Connect the wire from pin 4 to the momentary side of the toggle switch. This is now the load interrupt switch. Connect the lead from pin 64 to the other toggle. This will be the pause button. Now connect the wire from the ground on the board to the other sides of each of the three switches. This completes the modification. Route the wires out of the silver cover to the computer board through the slot on the bottom side, and reassemble the computer.



RESET: Use like the quit button (FCTN =).

PAUSE: Stops a program cold; release switch to continue.

INTERRUPT: Runs an assembly program in upper memory if one has been previously loaded, such as a screen dump routine. Press for only a half second or so.

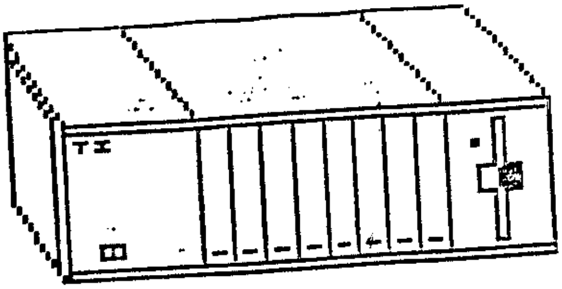
presented is liberally
ks, but at the risk of
tion is shown.
input while line 280
n. Line 290 builds the
ng.
the ASCII for "space"
ne 350 similarly counts
n of counting syllables
cts vowels and tentati-
is incremented to de-
Line 460 decrements the
consecutive vowels.
or "ted" and by jump-
ses 490 which rejects
ub-strings must include
nly concerned here with
510 similarly by-passes
being rejected under

sier. Line 550 detects
SCII for "space", tests
llables in the word and
he "long word" count.
syllable count for the
ests for 100 or more
for an adequate sample)
f the sentence.
the ASCII for "ENTER"
to break off the test-
o that you may test the
r curiosity. Line 620
letter. You will note
happening between one
If you are a speed
to slow down your pace.
tence has a clearly in-
should be treated as a
periods to break it

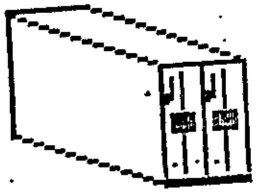
ood

```
       560 LW=LW+1
       570 IF C<>32 THEN 600
 VO    580 S=0
       590 REM STOPS AT END OF SENT
       ENCE WITH 100 OR MORE WORDS
       600 IF (W>=100)*(C=46)THEN 6
       30
 ED)   610 IF C=13 THEN 630
       620 GOTO 260
 " I   630 PRINT ::::::::"Number o
       f sentences =";NS:"Number of
       words =";W:"Words per senten
 " XI  ce =";W/NS
       640 PRINT "number of long wo
 " XI  rds =";LW
       650 F=INT((W/NS+100*LW/2)*.4
       +.5)
 3     660 PRINT :"FOG INDEX =";F
       670 END
 60
```

PBOX



by Bill Keller



DUAL DRIVE FULL POWER MODIFICATION

Have you ever found a good deal on a disk drive, and then found out that it was a full power drive and that you could not run it with your present P box. Well this little mod in your P box will enable you to run two (2) full power drives (either full heights - if you build a case for it to stand alone - , or two half height drives that will mount inside your P box).

If you attempt this modification you do so at your own risk. The mod is working in at least two (2) P boxes. This mod is brought to you with special thanks to Tony and Joe.

PARTS REQUIRED:

1 - +5volt REGULATOR 7805
1 - +12volt REGULATOR 7812
2 - 47uf 50volt CAPACITOR
2 - 3uf 25volt CAPACITOR
1 - CIRCUIT BOARD
1 - DISK DRIVE MALE SOCKET CONN.

#######################################
##################

Looking at the P box from the rear, select the pins on the Right Hand side of the motherboard socket. Pins one(1) and thirty(30) show a nominal +12 volts and +24 volts, again when you are looking from the rear of the P box. The grounds (and there must be two of them), are located at pins four(4) and twenty-four(24). If you find this hard to follow, get a digital multimeter and measure for +12v, +24v, and the two grounds and mark them as such. Next fashion your circuit board so that your pins will line up with the slots you have determinded are your hot and ground pins. The circuit board with completed runs should look something like figure #1. ##NOT DRAWN TO SIZE.##

By drilling little holes in your board the lead of your componets may be fastened easily and neatly to the board

The next thing to do is to soldier your two(2), 47uf capacitors to the board.. The capacitors should be connected: one lead to ground, and the other attached to one of your input slots. This should be done for both your +12v and +24v input slots. See Figure #3.

Next mount your regulators, remembering the 5v regulator goes with the +12v supply and 12v regualtor goes with the +24v supply. The regulators should have heat sinks, to keep them running cool. The pin out for the regulators are discribed in figure #2. Connect the input lead of your regulator to the same common point as your 47uf capacitor(Hot side). Connect output lead to your output slot. Soldier the ground to the respective ground on your board. See Figure #3.

The 3uf capicators, are connected from ground to the output slot of your board. See Figure #3.

A light can be rigged up to the front of your P box to tell you that in fact you have a circuit board running in that slot. I picked off the 12v line to run my lamp. See Figure #3.

The last thing to connect is your disk drive socket connector. By measuring your other drive's power line socket, you can tell which pins should be your +12v, +5v, and Ground leads for your new socket. The socket only -fits one way. Once you have the lines figured out then connect your wire to the corresponding power output slots from your board.

#####DO NOT GET YOUR 5V AND 12V LINES CROSSED OR YOU WILL DESTORY YOUR DISK DRIVE.#####

This little mod has eliminated some of my daul drive lockouts. I was running two(2) full power drives with just the single P box power supply. When I tried making back-up copies using both drives I would get read/write errors. Sometimes it would just initialize wrong, and sometimes the keyboard would lock me out. I now can run double density on the both drives(still have the old TI Controller) with no error.

Inside every small problem is a
larger problem struggling to get out!

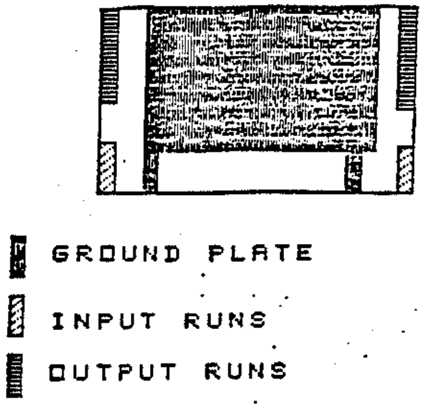


FIGURE #1

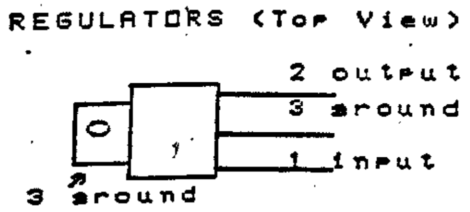


FIGURE #2

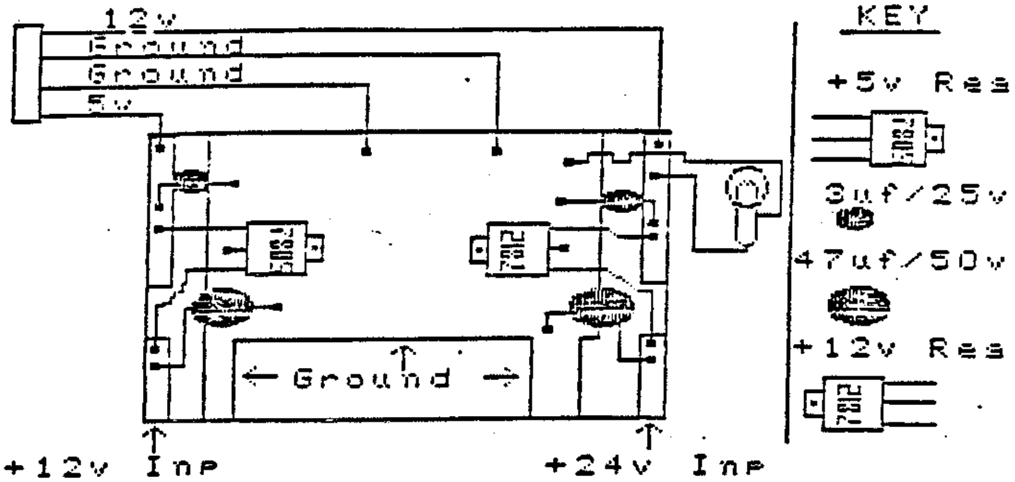


FIGURE #3

Erase All -- Day of the Week

We have two short programs for you this month that you may enjoy. The first program is called "Erase All". What it does is compare the result of using CALL CLEAR or ERASE ALL to clear the screen. Some say that ERASE ALL is quicker than CALL CLEAR but with this program there doesn't appear to be much difference.

Anyway try it out. It's simple to type in and you may improve your programming skills by studying the logic of the program.

The second program is entitled DAY/WEEK. With this program you can type in any date and the program will reveal to you what day of the week any event occurred on. To understand some of the logic of the program you might want to refer to our calendar program that we published in the January 1990 issue of Word Play. Adrian Johnson of the Orange County User Group provided an interesting history of how our present day calendar came about and the mathematics needed to research when a particular date occurred in the past.

Erase All

```
10 N=11
20 CALL CLEAR
30 DISPLAY AT(14,2):"<Press
<ANY> key to start>"
40 CALL KEY(0,K,S):: IF S=0
THEN 40
50 FOR I=1 TO N
60 CALL HCHAR(1,1,I+32,768)
70 CALL CLEAR
80 NEXT I
90 DISPLAY AT(14,10)BEEP:"TE
ST #1"
100 DISPLAY AT(14,2):"Press
<ANY> key to start>"
110 CALL KEY(0,K,S):: IF S=0
 THEN 110
120 FOR I=1 TO N
130 CALL HCHAR(1,1,I+32,768)
140 DISPLAY ERASE ALL
150 NEXT I
160 DISPLAY AT(14,10)BEEP:"T
EST #2"
170 GOTO 40
```

Day of the Week

```
100 INPUT "ENTER MM,DD,YYYY:
":M,D,Y
110 A=Y-(INT(Y/28)*28):: B=A
/4 :: E=A-INT(B)*7
120 C$="511462403513" :: IF
E=0 THEN IF M<3 THEN C$="40"
130 E=VAL(SEG$(C$,M,1)):: IF
 Y<1900 THEN A=A+12
140 G=A+INT(B)+D+E :: F=G-(I
NT(G/7)*7)
150 DATA SUN,MON,TUES,WEDNES
,THURS,FRI,SATUR
160 RESTORE :: FOR B=0 TO F
:: READ C$ :: NEXT B
170 PRINT "THE DAY IS ";C$;"
DAY"
180 PRINT
190 INPUT "DO ANOTHER? (Y/N)
: ":YN$
200 IF YN$="Y" OR YN$="y" TH
EN GOTO 100
```

Root Finder

Besides having the ability to play great mes and run many other complex programs, ur TI-99/4A is also a superior number crunching machine. The following 2 programs demstrate its ability to find the square root 9th root of any number up to ten digits.

The first program will find the root and splay its calculations on the screen as it lves the problem. The second program is ch faster, reveals an almost instantaneous answer, but does not show the calculations the screen.

I checked the answers on several test numbers with Macmillan's Logarithmic and Trigonometric Tables and the answers are absolutely correct. This book, by the way, is what I had to use when I was a college math student many years ago. What a difference today!

--Charles Ball, Editor

Root Finder with display

```
90 CALL CHARPAT(121,CH$):: C
ALL CHAR(33,CH$)
100 DISPLAY AT(1,1)ERASE ALL
:"  ROOTS b! Lucie Dorais":"
To find an! root from cube
root to 9th root"
110 !
120 ON WARNING NEXT :: PR$="
PIO"
130 L$=RPT$("}",28):: E$=RPT
$(" ",168):: S$=RPT$(" ",8)
140 CALL CHAR(120,"000000000
002050F",121,"1F102020404080
80",122,"018182C2C4646830",1
23,"0B0101",125,"FF")
150 DISPLAY AT(5,9):"x3y"&RP
T$("}",10):S$&"{z" :: GOSUB
280
160 ACCEPT AT(6,12)VALIDATE(
NUMERIC)BEEP:N :: IF R>2 THE
N 180
170 IF R=1 THEN AV=N :: GOTO
 220 ELSE AV=SQR(N):: GOTO 2
20
180 LO=0 :: HI=SQR(N)
190 AV=(LO+HI)/2 :: T=AV^R
200 IF OAV=AV THEN 220 ELSE
OAV=AV :: DISPLAY AT(12,8):A
V
210 IF N<T THEN HI=AV :: GOT
O 190 ELSE IF N>T THEN LO=AV
 :: GOTO 190
220 AN=AV :: DISPLAY AT(12,8
)BEEP:"=";AN
230 DISPLAY AT(22,1):L$:" {A
}nother   {C}hange root   {P
}rint     {Q}uit"
240 CALL KEY(0,K,S):: IF S=0
 THEN 240 ELSE K=POS("ACPQ",
CHR$(K),1)
250 IF K=0 THEN 240 ELSE ON
K GOTO 260,260,270,290
260 DISPLAY AT(7,12):E$:E$:E
$ :: IF K=2 THEN GOSUB 280 :
: GOTO 160 ELSE 160
270 OPEN #1:PR$ :: PRINT #1:
S$&"_______________":S$&" "&ST
R$(R)&"/" :: PRINT #1:S$&"\/
 ";N;TAB(26);"=";AN:"" :: CL
OSE #1 :: GOTO 240
280 ACCEPT AT(5,10)VALIDATE(
"123456789")SIZE(-1)BEEP:R :
: RETURN
```

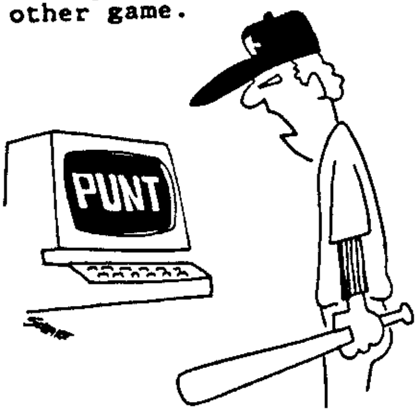
Root Finder

```
90 CALL CHARPAT(121,CH$):: C
ALL CHAR(33,CH$)
100 DISPLAY AT(1,1)ERASE ALL
:"  ROOTS b! Lucie Dorais":"
To find an! root from cube
root to 9th root"
110 !REVISED VERSION USINT T
HE FORMULA AN=N^(1/R)
120 ON WARNING NEXT :: PR$="
PIO"
130 L$=RPT$("}",28):: E$=RPT
$(" ",168):: S$=RPT$(" ",8)
140 CALL CHAR(120,"000000000
002050F",121,"1F102020404080
80",122,"018182C2C4646830",1
23,"0B0101",125,"FF")
150 DISPLAY AT(5,9):"x3y"&RP
T$("}",10):S$&"{z" :: GOSUB
280
160 ACCEPT AT(6,12)VALIDATE(
NUMERIC)BEEP:N :: IF R>2 THE
N 180
170 IF R=1 THEN AV=N :: GOTO
 220 ELSE AV=SQR(N):: GOTO 2
20
220 DISPLAY AT(12,8)BEEP:"="
;AN
230 DISPLAY AT(22,1):L$:" {A
}nother   {C}hange root   {P
}rint     {Q}uit"
240 CALL KEY(0,K,S):: IF S=0
 THEN 240 ELSE K=POS("ACPQ",
CHR$(K),1)
250 IF K=0 THEN 240 ELSE ON
K GOTO 260,260,270,290
260 DISPLAY AT(7,12):E$:E$:E
$ :: IF K=2 THEN GOSUB 280 :
: GOTO 160 ELSE 160
270 OPEN #1:PR$ :: PRINT #1:
S$&"_______________":S$&" "&ST
R$(R)&"/" :: PRINT #1:S$&"\/
 ";N;TAB(26);"=";AN:"" :: CL
OSE #1 :: GOTO 240
280 ACCEPT AT(5,10)VALIDATE(
"123456789")SIZE(-1)BEEP:R :
: RETURN
```

The Wave

Stand up and cheer for your team. "Wave" emulates the crowd at some etic event. When you run the program you'll see that fans in the bleachers stand and then sit. This is reted over and over again and will ind you of the spectators at the zer or some other game.

```
90 !THE WAVE by David Renken
berger/mod by Jim Peterson
100 CALL CLEAR :: CALL SCREE
N(4)
110 A$="xxthe wavexx"
120 DISPLAY AT(4,14-LEN(A$)/
2):A$
130 B$="press any key to sto
p"
140 DISPLAY AT(22,14-LEN(B$)
/2):B$
150 B$="005A3C3C3C3C2466"
160 A$="000018187EB03C3C"
170 FOR CH=91 TO 118 :: CALL
 CHAR(CH,A$):: M$=M$&CHR$(CH
):: NEXT CH :: FOR R=8 TO 12
 :: DISPLAY AT(R,1):M$ :: NE
XT R
175 FOR T=1 TO 26 STEP 5 ::
DISPLAY AT(22,T):SEG$(M$,T,1
):: NEXT T
180 FOR CH=91 TO 123 :: CALL
 CHAR(CH,B$):: CALL CHAR(CH-
5,A$):: CALL SOUND(-999,-7,5
*RND):: CALL KEY(3,K,ST):: I
F ST<>0 THEN CALL HCH :: STO
P
190 NEXT CH :: GOTO 180
200 SUB HCH :: CALL HCHAR(8,
1,31,160):: CALL HCHAR(22,1,
32,32):: SUBEND :: END
```



. . . Wrong Season!

o get a Horizon Ram Disk

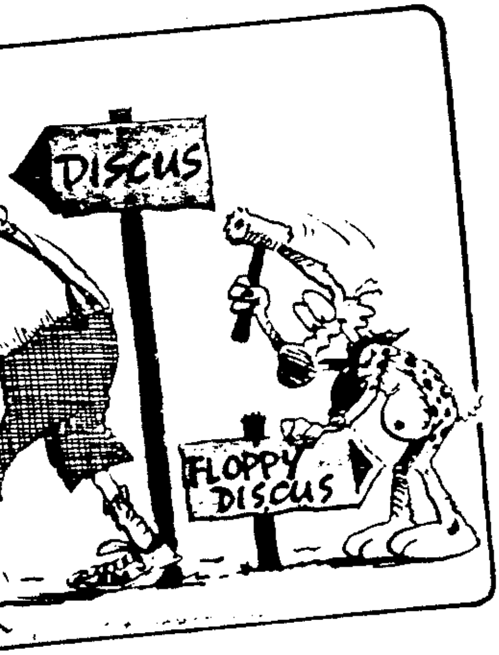
witch on the card to #5
o call the ram disk DSK6
to call the ram disk
ch is set in the "closed"

only. Boot use to DOS.
am CONFIG-16 which is on
Most people will not give
u buy the Ram Disk so you
copy from someone such as

ment "ASSIGN F=DSK6:" or
your AUTOEXEC files.
1000 and initalize the ram
u want.
iles into the ram disk.
the ram disk with any MDOS

uctions are for the Horizon
Disk and may not work with

help feel free to call.
Wicklund, Neville Blair and
eir help. Without them the
be on it's way back to the



We Buy, Sell and Trade...

USED TI-99/4a HARDWARE and SOFTWARE

Looking to sell your old TI-99/4a equipment? Need some extra cash? Interested in picking up a second system for the kids, the office, or as a backup for your existing equipment? If you answered yes to any of these questions you need to contact Texaments! We buy, sell and even trade used TI-99/4a hardware, software, command modules and accessories. We buy working and non-working equipment. And we guarantee that anything you buy from us is in perfect working order... or your money back (unless noted otherwise). No gimmicks and no surprises. All items listed will be sold on a first come first serve basis only. For an up-to-the-minute listing of used items please call TI Source(sm), our 24 hour multiuser bulletin board system, at (516)475-6463 (A modem is required to call the TI Source(sm) bulletin board system)

If you're interested in selling some or all of your existing TI-99/4a gear please call or write us immediately. Those contacting us by mail will be sent a formal offer (including acceptance instructions) by return mail within 2 business days upon receipt of your letter. This quotation service is free and you may reject our offer by simply not replying to it. It's that simple!

Hardware (Computers, Peripherals & Accessories)

TI-99/4a Console (Black & Silver)	$22.95 *(7.00)*
TI Peripheral Expansion Box	$79.95 *(18.00)*
TI Disk Controller Card	$29.95
TI Peripheral Expansion Interface Card	$39.95
TI SS/SD Disk Drive	$14.95 *(5.00)*
TI Cassette Deck Cable	$2.95
TI Dual Cassette Deck Cable	$3.95
TI Impact Printer (Epson MX-80)	$69.95 *(8.50)*
Simth Corona Fastext 80 Printer with parallel cable & 2 ribbon cartridges	$29.95 *(8.50)*
TI-99/4a Video Modulator (with Composite RCA-jack output)	$3.95
TI-99/4a Composite Video Cable (New)	$12.95
Module "L" Connector Replacement Part (New – fits inside TI-99/4a Console)	$7.95
PointMaster Arcade Style Joystick with TI Wico Adapter	$9.95
•Tandon TM-1002A Full Height DS/DD Floppy Drive	$10.00
Percom Data Expansion Drive Unit (with power supply and cable – built like a tank! – no docs, no guarantee)	$17.50
IBM Keyboard Interface Prototype (attaches to joystick port – from Germany, no docs, not sure if it works, no guarantee)	$3.95
8-Foot Hayes Modem Cable (round type cord -- no operational guarantee)	$4.95
Myarc DS/DD Controller Card (in classic metal enclosure -- overheats, needs repair, no guarantee)	$35.00
Dual 1/2-Height Floppy Enclosure with Power Supply	$29.95
Defective (Used) 1/2-Height DS/DD Floppy Drive (no operational guarantee)	$2.50

Software (Modules, Disk & Cassette)

Key: (M) = Module/Solid State Cartridge, (C) = Cassette, (D) = Disk

Module Special: Get any 2 modules priced at $3.95 or less for only $5.95, any 3 for only $9.95, or any 5 for $15.95!

4a Flyer Flight Simulator (M)	$9.95
TI Writer (M/D)	$9.95
Myarc Extended Basic II (Module only - No disk or documentation)	$7.95
Adventure Module (with Pirate Adventure) (M/D)	$3.95
Parsec (M)	$3.95
Munchman (M)	$3.95
Super Demon Attack (M)	$3.95
Picnic Paranoia (by Atari) (M)	$4.95
Personal Record Keeping (M)	$4.95
Household Budget Management (M)	$3.95
Hunt The Wumpus (M)	$3.95
Tombstone City (M)	$3.95
Donkey Kong (by Atari) (M)	$4.95
Teach Yourself Basic (C)	$2.95
Terminal Emulator II (M)	$5.95
Tunnels of Doom (M/C)	$3.95
Tunnels of Doom (Module only; cassette missing)	$1.95
Home Financial Decisions (M)	$3.95
Tax Investment/Record Keeping (M)	$3.95
Business Manager (Invoice, Mail List, Order Entry) (D) by Yu-Can Software	$4.95
Mini Memory (M)	$9.95

Books, Manuals & Miscellaneous Items

Sorry... books may not be returned. The author or authors of each book are noted in (parenthesis).

Systemate 99/4a Console Cover (clear plastic cover that fits perfectly over the console)	$1.95
TI-99/4a Users Reference Guide (TI)	$1.95
Compute's First Book of TI Games (Compute!)	$3.95
Beginners Basic (TI)	$1.95
Programs for TI Home Computer (Steve Davis)	$4.95
Editor/Assember Manual (TI)	$2.95
The TI-99/4a Users Guide (Casciato & Horseall)	$2.95
CorComp 9900 Disk Controller Manual (CorComp)	$1.50

TEXAMENTS

Serving the TI Community Since 1984

Office: 516-475-3480 53 Center Street, Patchogue, NY 11772 BBS: 516-475-6463

Please add $3.25 for domestic first class and Canadian delivery, $8.50 for foreign air mail delivery. Orders are typically shipped within 48 hours of receipt. C.O.D. orders must be placed by phone. We do not accept credit cards. For more information please call or write.